

PRÁCTICA 2

ÁRBOL DE DECISIÓN (ID3)



INGENIERÍA DEL CONOCIMIENTO

Jaime Jiménez Nieto

Introducción

El objetivo de la práctica 2 de la asignatura Ingeniería del Conocimiento es implementar una versión simplificada del algoritmo del árbol de decisión (ID3) visto en las clases teóricas.

A continuación, se explica de manera breve el contenido de los siguientes apartados:

- **Detalles de la implementación:** Se explica el lenguaje de programación utilizado, así como la estructura de carpetas seguida.
- **Manual de usuario:** Se realiza una explicación detallada de cómo replicar la aplicación en el sistema del usuario de manera local.
- **Ejemplos:** Se muestra una serie de ejemplos a modo de prueba de la aplicación.

Detalles de la implementación

El algoritmo A* se ha desarrollado utilizando **Typescript** y para conseguir la visualización se ha utilizado **Angular** como framework. Todo ello implementado en Visual Studio Code como IDE.

Estructura del proyecto

Lógica de negocio: Se han creado los siguientes ficheros para implementar la lógica de la aplicación:

- **example.ts:** donde se especifica para cada atributo su valor. Esta clase tendrá los siguientes atributos:
 - **_attributes:** map que almacena los atributos junto con sus valores asociados.
- **dataset.ts:** donde se permite manipular y acceder a los diferentes conjuntos de datos utilizados en el algoritmo. Esta clase tendrá los siguientes atributos:
 - **_examples:** ejemplos que representan los conjuntos de datos.
 - **_attributes:** nombres de los distintos atributos de los conjuntos de datos (sin contar con el atributo del resultado).
 - **_resultName:** atributo del resultado del conjunto de datos.
- **node.ts:** clase que representa un nodo del árbol de decisión cuyas propiedades son las siguientes:
 - **_subtable:** Es un objeto Dataset que representa el conjunto de datos asociado a este nodo.
 - **_isLeaf:** Es un booleano que indica si este nodo es una hoja en el árbol de decisión.
 - **_children:** Es un array de objetos Node que representan los hijos de este nodo en el árbol.
 - **_value:** Es un string opcional que representa el valor asignado a este nodo si es una hoja.
 - **_fatherValue:** Es un string que representa el valor del atributo del padre de este nodo en el árbol.
 - **_attributes:** Es un array de objetos Attribute que representan los atributos asociados al conjunto de datos de este nodo.

- **_bestAttribute:** Es un objeto Attribute que representa el mejor atributo para dividir el conjunto de datos en este nodo.
- **attribute.ts:** clase que proporciona funcionalidades para calcular el mérito de un atributo en un conjunto de datos, lo que facilita la selección del mejor atributo para dividir los datos en un árbol de decisión durante el entrenamiento de un modelo de aprendizaje automático. Las propiedades son las siguientes:
 - **_name:** Es un string que representa el nombre del atributo.
 - **_positives:** Es un mapa que almacena la cuenta de valores positivos (por ejemplo, "sí") para cada valor del atributo.
 - **_values:** Es un mapa que almacena la cuenta total de ocurrencias de cada valor del atributo.
 - **_numberExamples:** Es un número entero que representa el número total de ejemplos en el conjunto de datos.
 - **_merit:** Es un número que representa el mérito del atributo para la división en el árbol de decisión.
- **id3.ts:** fichero que proporciona el método principal (o main) del algoritmo junto con otras funciones para hallar las diferentes reglas.

Visualización: Para la visualización ha sido falta la creación de varios componentes de Angular, cada uno de ellos ejecutando una única función:

- **file-reader:** componente que se encarga de realizar la carga de los diferentes datos a partir de un archivo.
- **board:** componente que se encarga de realizar la llamada al algoritmo una vez los datos se encuentren cargados y listos.

Manual de usuario

Para poder ejecutar el proyecto es necesario tener instalado Node.js y Angular en el sistema. Node.js se puede instalar a través de su página web: <https://nodejs.org/en>. Una vez instalado Node.js se ejecuta el comando: `npm install -g @angular/cli` para instalar Angular

1. Importar el proyecto ID3 en Visual Studio Code

2. Instalación de los paquetes necesarios

Para poder ejecutar el proyecto es necesario instalar todos los paquetes declarados en el fichero package.json. Para ello se utiliza el comando **npm i**.

3. Arrancar el proyecto

Se ejecuta en la terminal del proyecto el comando **npm start** y se abre en un navegador predeterminado la dirección localhost:4200 para poder acceder a la aplicación. El resultado es el siguiente:

Árbol de decisión ID3

Atributos

Choose File no file selected

Ejemplos

Choose File no file selected

Resolver

4. Carga de los datos

Se realiza la carga de los diferentes conjuntos de datos, tanto de los atributos como de los ejemplos a partir de los diferentes botones ubicados en la parte derecha de la pantalla

Atributos

Choose File



AtributosJuego.txt

Ejemplos

Choose File



Juego.txt

Resolver

5. Resolver el problema

Se hará click en el botón **Resolver** para que la aplicación aplique el algoritmo de decisión a partir de los datos recibidos. El resultado será el conjunto de reglas calculadas, a partir de realizar un recorrido en profundidad por el árbol:

Árbol de decisión ID3

Si TiempoExterior = soleado y Temperatura = caluroso entonces Jugar = no
Si Temperatura = templado y Humedad = alta entonces Jugar = no
Si Humedad = normal entonces Jugar = si
Si Temperatura = frio entonces Jugar = si
Si TiempoExterior = nublado entonces Jugar = si
Si TiempoExterior = lluvioso y Temperatura = templado y Humedad = alta y Viento = falso entonces Jugar = si
Si Viento = verdad entonces Jugar = no
Si Humedad = normal entonces Jugar = si
Si Temperatura = frio y Humedad = normal y Viento = falso entonces Jugar = si
Si Viento = verdad entonces Jugar = no

Atributos

Choose File



AtributosJuego.txt

Ejemplos

Choose File



Juego.txt

Resolver

Ejemplos

Ejemplo 1

Utilizando el conjunto de datos proporcionado en la práctica se obtiene el siguiente resultado:

Árbol de decisión ID3

Si TiempoExterior = soleado y Temperatura = caluroso entonces Jugar = no
Si Temperatura = templado y Humedad = alta entonces Jugar = no
Si Humedad = normal entonces Jugar = si
Si Temperatura = frio entonces Jugar = si
Si TiempoExterior = nublado entonces Jugar = si
Si TiempoExterior = lluvioso y Temperatura = templado y Humedad = alta y Viento = falso entonces Jugar = si
Si Viento = verdad entonces Jugar = no
Si Humedad = normal entonces Jugar = si
Si Temperatura = frio y Humedad = normal y Viento = falso entonces Jugar = si
Si Viento = verdad entonces Jugar = no

Atributos

Choose File



AtributosJuego.txt

Ejemplos

Choose File



Juego.txt

Resolver

Ejemplo 2

Utilizando el conjunto de datos tomados en las diapositivas de teoría (diapositiva 18)

Árbol de decisión ID3

Si tamaño = pequeño y timbres = uno entonces clase = si
Si timbres = varios entonces clase = no
Si tamaño = mediano entonces clase = si
Si tamaño = grande entonces clase = no

Atributos

Choose File



AtributosJuego2.txt

Ejemplos

Choose File



Juego2.txt

Resolver

Ejemplo 3

Utilizando los ficheros AtributosJuego3 y Juego3 ubicados en la carpeta Assets de la práctica:

Árbol de decisión ID3

Si Edad = Joven y NivelEducativo = Bachillerato y ExperienciaLaboral = 2 años entonces EstadoCivil = Soltero
Si ExperienciaLaboral = 3 años entonces EstadoCivil = Casado
Si NivelEducativo = Universitario entonces EstadoCivil = Soltero
Si Edad = Mediana y NivelEducativo = Bachillerato entonces EstadoCivil = Soltero
Si NivelEducativo = Universitario y ExperienciaLaboral = 2 años entonces EstadoCivil = Casado
Si ExperienciaLaboral = 3 años entonces EstadoCivil = Soltero
Si Edad = Mayor entonces EstadoCivil = Casado
Si Edad = entonces EstadoCivil = undefined

Atributos

Choose File



AtributosJuego3.txt

Ejemplos

Choose File



Juego3.txt

Resolver